

NEXT GENERATION JAVASCRIPT

Att modernisera en legacy-kodbas

CALLE BOKETOFT

- Webbutvecklare - front end lead Lime CRM
- Pluggat Interaktionsteknik & Design i Umeå
- Många typer av projekt, nyskrivning - legacy
- Single page apps - administrationsverktyg
- Nu: Lime CRM - modernisering av webbplattform



- Funnits sedan 90
- Växer och anställer för fullt
- Lund, Stockholm, Göteborg
- Produktföretag - Säljverktyg

ÖVERSIKT

1. Bakgrund
2. Analys - existerande
3. Analys - framtid
4. Planering
5. Utförande
6. Risker
7. Nästa steg?
8. Våra valda lösningar

1. BAKGRUND FÖR CASET

- Pågående projekt sedan ett antal år
- Medelstor/stor kodbas
- Prio nya features och buggfix
- Teknisk skuld påtaglig, starkt motiverande
- Önskescenario: modern JS-kodbas
- Börja om på nytt? Modernisera?

2. ANALYS - EXISTERANDE KOD

- Kolla igenom koden på utvalda ställen
- Få översikt över de olika delarna
- Automatiserade tester?
- Vilket byggsystem används?
- Är koden modulariserad?
- Serverrendering eller klientrendering?
- Hemskrivna lösningar?

3. ANALYS - FRAMTIDA KOD

- Mantra: förenkla!
- Vilka browsers ska vi stödja?
- Vad kan tas bort?
- Modularisera delar som bör delas upp
- Sikta på enklast möjliga byggsystem
- Standardiserad kodformattering

4. PLANERING

- Sätt upp tydliga mål
- Avgränsade iterationer
- Finns en naturlig ordning av mål?
 - Mycket nytta med lite arbete i början
- Ungefärlig tidsplan per mål
 - Fastnar man kan man omorganisera

5. UTFÖRANDE AV ARBETE

- Börja med byggsystem, enklast att verifiera
- Logisk och lättnavigerad mappstruktur
- Modularisera med minimal förändring
- Samma beteende som tidigare helst
- Kom överens om nya beteenden i teamet
- Testa regelbundet - både auto samt manuellt
- Alltid en sak åt gången!

6. RISKER

- Göra mer än en sak åt gången
 - Upptäcker något fel senare - backtracking
 - Rena GIT commits viktigt
- Ersätta en komplex lösning med en annan
- Byta ut något "bara för att"

7. NÄSTA STEG?

- Kodens grund och struktur är redo
 - Kodförbättring, nyutveckling
- Större uppdateringar
 - AngularJS -> Angular 5
- State management med Redux
- Web Components
 - Fristående UI-komponenter

VÅRA VALDA LÖSNINGAR



JAVASCRIPT-MODULER



- Ny standard
 - Föregångare AMD, CommonJS
- Dela upp i explicita moduler
- Kodnavigering i editor
- Autocomplete, intellisense

WEBPACK



- Automatiserat bygge av JS
- Bygga ihop JS moduler för browser
 - Även CSS-moduler etc.
- Live reload - Automatisk omladdning
- Cache busting, minifiering

COMPILE TO JAVASCRIPT



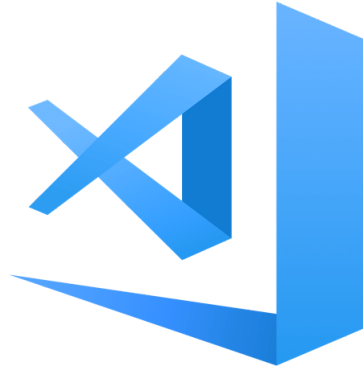
- TypeScript: Superset av JavaScript
 - Kommande standarder idag
 - Optional types
 - Underlättar arbete i team
 - Enkelt med refaktorering

DATA STORE



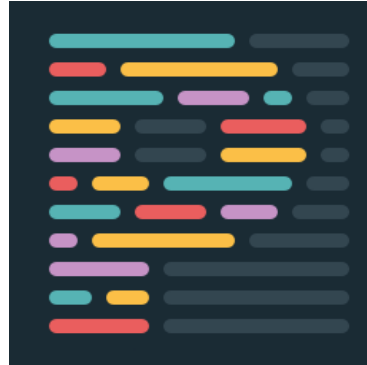
- Data hanteras på ett ställe
- Enkelt att organisera och testa
- Uppdateringar till alla konsumenter
- Time traveling - "undo"

EDITOR MED KODANALYS



- Visual Studio Code
- Atom
- VIM
- Autocompletion etc.
- Förstår TypeScript

KODFORMATTERING



- Prettier.js
- Opinionated
- Utvalda konfigureringar möjliga
- Automatiskt formatterning
 - Skrämmande tills man testat

FRONT END FRAMEWORK



- Viktigt val för hela produkten
- Komplette FE ramverk för team
- Components för UI element
- FormBuilder, Router, HTTP Client
- Observables, One way binding

TACK!

- Traineeprogram för nyutexaminerade
 - lundalogik.se/trainee
 - Sista ansökningsdag 28 februari
- Maila oss info@lundalogik.se
- github.com/Lundalogik/next-gen-js-slides